GOL Reference Manual

Gigabit Optical Link
Transmitter manual

P. Moreira*, T. Toifl, A. Kluge, G. Cervelli,
A. Marchioro, and J. Christiansen

Preliminary-

CERN - EP/MIC, Geneva Switzerland

March 2001

Version 0.1

*Technical contact person e-mail: Paulo.Moreira@cern.ch

Summary of changes

Version 0.1

Pin names corrected for bond-pads number 13 to 16. Please see pin assignments table on page 31.

Chapter 1

Introduction

Gbit/s data transmission links will be used in the in trigger and data acquisition systems of several LHC detectors. In these applications, the transmitters, located inside the detectors, will be subject to high radiation doses.

The Gigabit Optical Link (GOL) chip, is a multi-protocol high-speed transmitter ASIC, which is able to withstand high doses of radiation. The IC supports two standard protocols, the G-Link and the Gbit-Ethernet, and sustains transmission of data at both 800 Mbit/s and 1.6 Gbit/s. The ASIC was implemented in a 0.25 µm CMOS technology employing radiation tolerant layout practices.

SYSTEM OVERVIEW

In the four LHC experiments (ATLAS, CMS, ALICE and LHCb), high-speed (Gbit/s) data links will be used in the trigger and data acquisition systems. In these links, the flow of information will be unidirectional, that is, the transmitters will be located inside the detectors and the receivers will be situated in the experiment's counting rooms. A consequence of this arrangement is that the transmitters will be subjected to high levels of radiation while the receivers will operate in a radiation free environment.

In this manual, the Gigabit Optical Link (GOL) transmitter ASIC, that has been designed to operate reliably under the radiation conditions encountered inside the LHC detectors, is described. Since the receivers do not require any type of radiation hardness, commercial devices will be used together with GOL IC's to assemble complete data links.

Operating the transmitter ASIC in a link with a standard commercial receiver imposes some compatibility constraints: Namely, data formats, data rates and coding schemes have to be respected. Additionally, for trigger links the constant latency requirement imposes data rates that are multiples of the LHC master clock frequency. In most applications, the detector systems require the transmission of either 16 or 32 bits of data in a single LHC clock cycle. The ASIC was designed to support both the 8B/10B [1] and the CIMT [2] line coding schemes. Both schemes introduce an overhead of two additional bits for each eight bits of data. Therefore, the required data rates are either 800 Mbit/s or 1.6 Gbit/s for 16 or 32-bit data transmission. These result in effective data bandwidths of 640 Mbit/s and 1.28 Gbit/s, respectively.

GOL Architecture

The block diagram of the ASIC is shown in Figure 1. Its operation can be described as follows. At every master clock cycle (LHC clock), data is presented to the transmitter inputs either as a 16 or 32-bit word. When the 32-bit mode is selected the input data is time division multiplexed in two 16-bit words that are sequentially processed by the line-code encoders. If 16-bit mode is selected, only a single word is processed in a 40.08 MHz master clock cycle. Which line-coding scheme is used is the choice of the user. If the "Conditional-Invert Master Transition" (CIMT) [2] encoding scheme is employed, a G-Link receiver is required while if, 8B/10B [1] coding is performed then either Gbit Ethernet or Fibre Channel receivers can be

used, provided that they are compatible with the data rate being generated. While CIMT encoding is done in a group of 16-bits at a time, 8B/10B encoding is specified for 8-bit words. To avoid treating the two coding schemes differently, the 8B/10B encoder was designed to process two 8-bit words in parallel. Before being fed to the serializer, the 20-bit encoded words are time division multiplexed in two 10-bit words by the word-multiplexer. The 10-bit high-speed serializer converts its input into a serial stream of either 800 Mbit/s or 1.6 Gbit/s. The serial data is then fed to the laser-driver and to the 50 Ω line-driver. These can be used either to intensity modulate a laser or to drive a 50 Ω transmission line with Pseudo ECL levels. Due to radiation effects, it is expected that the threshold current of the laser diodes will increase with time over the lifetime of the experiments [3]. To compensate for this, the laser-driver contains an internal pre-bias current generator that can be programmed to sink currents between 0 and 55 mA. Programming the ASIC can be done using either an I2C [4] or a JTAG [5] interface. External hardwired pins set the main operation modes of the receiver.

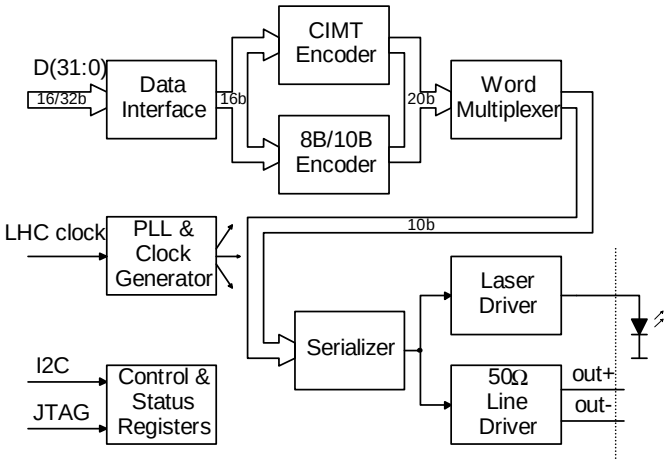

Figure 1 GOL block diagram.

Modes of operation

Mode	line coding scheme	**conf_ wmode16** pin	**conf_ glink** pin	Input bus width	Internal clock speed [MHz]	data rate [Mbit/s]	net data rate [Mbit/s]
Ethernet fast	8B/10B	0	0	32	80	1600	1280
G-LINK fast	CIMT	0	1	32	80	1600	1280
Ethernet slow	8B/10B	1	0	16	40	800	640
G-LINK slow	CIMT	1	1	16	40	800	640

Table 1 The four basic modes of operation

The Gigabit Optical Link (GOL) transmitter circuit supports a total of 2 x 2 different modes of operation: **G-Link** and **Ethernet** mode, both either **fast** (*1.6 Gbit/s*) or **slow** (*0.8 Gbit/s*). These modes, which are summarised in Table 1, are selected by wiring the pins *conf_glink* and *conf_wmode16* to the required logic value. A logic “1” on

conf_glink selects G-Link mode. Otherwise, the chip operates in Ethernet mode. A logic “1” on *conf_wmode16* selects the slow transmission mode, using only the 16 least significant bits of the input data bus (*din<15:0>)*. A logic “0” chooses fast transmission mode, using 32-bit data, (*din<31:0>*). In both G-Link and Ethernet mode, the line coding adds 2 bits for each 8 bits of data, such that the effective data rate is 80 % of the transmission bandwidth. The encoding takes care that the transmitted sequence is DC free, and the run-length of zeros or ones is limited.

Fast and slow transmission

In the *fast* transmission mode, which is selected when *conf_wmode16* is zero, 32 bits of data are transmitted during a 40 MHz clock cycle, resulting in 1.6 Gbit/s data rate. In the *slow* transmission mode, only 16 bits of data are transmitted, resulting in a final data rate of 800 Mbit/s. Throughout this manual, the denominations "*fast*" and "*32-bit*" mode, and "*slow*" and "*16-bit*" mode will be used synonymously.

G-Link mode

When the chip is configured to be in G-Link mode, the “Conditional-Invert Master Transition” (CIMT) [2] transmission format is used. The chip reads the data to be transmitted from the input bus *din<31:0>*. The pins *cav*, *dav* and *force_ff0*, are used to control the transmission. An additional "flag" data bit can be added to each 16-bit word via the *flag<1:0>* signals. *flag<1>* is transmitted together with the upper 16 bits (corresponding to *din<31:16>*), and *flag<0>* to the lower sixteen bits (*din<15:0>*). In the 16-bit mode, only bits din<15:0> and *flag<0>* are used.

If the chip is in the “READY” state, the value on *din<31:0>* is transmitted as either a data or a control word, depending on whether the *dav* or the *cav* signal is activated. In the case that neither *cav* nor *dav* is “1”, the idle pattern is transmitted. If *force_ff0* is “1”, then the special symbol *ff0* is sent, which allows the receiver to gain synchronisation (see Table 2). In the 32-bit mode, data is transmitted with the lower bits first, i.e. the data word *din<15:0>* is serialised before *din<31:16>*. When transmitting a control frame, only bits din<39:16> and din<13:0> are used (two 14-bit control words). In the 16-bit mode, only bits din<13:0> are used (one 14-bit control word).

cav	*dav*	force_ff0	**G-Link Frame**
X	X	1	IDLE (ff0)
0	0	0	IDLE
0	1	0	DATA, din<31:0>
1	0	0	COMMAND frame
1	1	0	COMMAND frame

Table 2 G-Link transmission modes

Ethernet mode

In Ethernet mode, each byte is converted into a 10-bit word for transmission using the 8B/10B-encoding standard [1]. The transmission order is defined as going from the lower to the upper bits. Thus, *din<7:0>* is transmitted before *din<15:8>* etc. In Ethernet mode, two signals control the transmission: *tx_en*, and *tx_er*. These pins are shared with *dav* and *cav* used for G-Link mode. If the chip is in the “READY” state, the symbols will be transmitted according to Table 3. When *tx_en* is asserted and *tx_er* is not asserted then the data bits from pins *din<31:0>* are encoded and transmitted normally. When *tx_en* is not asserted and *tx_er* is asserted, then the

encoder will generate a carrier extend [1] consisting of four (or two[1]) <K23.7> codes. If *tx_en* and *tx_er* are both asserted then the transmit error propagation code (<K30.7>) is generated. During initialisation and after the chip loses synchronization, the IDLE sequence (<K28.5>, D5.6> or <K28.5>, <D16.2>, see [1]) is transmitted.

tx_en	*tx_er*	**Encoded 10 bit output**
0	0	IDLE (<K28.5>, D5.6>, <K28.5>, <D16.2>)
0	1	Carrier extend (<K23.7>)
1	0	Normal data (from din<31:0>)
1	1	Transmit error propagation (<K30.7>)

Table 3 Ethernet transmission modes.

[1] Depending on the transmission mode: Four bytes in fast (32-bit), 2 bytes in slow (16-bit) mode.

Chapter 2

Internal Registers

In this Chapter, the internal registers of the gigabit optical link, accessible via the I2C and the JTAG interface, are described in detail. A brief summary of this chapter can be found in Appendix B.

GOL REGISTER FILE

The GOL chip contains six user-accessible registers, which are listed in Table 4. The register file is divided into four configuration registers and two status registers. While configuration registers can be both read and written, status registers can only be read. The configuration registers are internally protected with Hamming check bits. If one bit flips due to a single event upset (SEU), it is automatically corrected (see also Chapter 7, "Measures against radiation effects").

I2C reg. address	Register name	Default content (after reset)
Configuration registers		
0	Config 0	00110011 ($33)
1	Config 1	00011111 ($1F)
2	Config 2	00010000 ($10)
3	Config 3	00100000 ($20)
Status registers		
4	Status 0	00000000 ($00)
5	Status 1	-

Table 4. The GOL register file.

Register access via the I2C bus and JTAG interface

All four configuration registers can be read and written via both the I2C and the JTAG interface. The two status registers allow only read access. How to use the I2C and the JTAG interfaces to access these registers is described in Chapter 3 and Chapter 4, respectively.

Setting parameters via registers or pins

Some parameters, i.e. the PLL charge pump current, the laser driver current and the line driver strength can either be chosen by hard-wiring pins (e.g. *conf_i_pll<4:0>*) or by using the values stored in the configuration register. Bit 7 of Config register 3 ("*use_conf_regs*") determines where the settings should be read from the pins ("0") or the registers ("1"). The "pin" setting is intended for a usage of the GOL where register access with either the I2C or JTAG bus is not possible. The default value of *use_conf_regs* is "0", thus after a reset the settings from the pins are used. Since the number of pins was limited, the hard-wired settings only decode a subset of the register settings. Table 20, 18, and 19 (page 41) define the pin encoding, and their corresponding register settings.

CONFIGURATION REGISTERS

Config 0

Bits	Name	Description
<4:0>	Wait_time	Defines the time to wait between the "LOCKED" state and "READY" state
<7:5>	Loss_of_lock_time	Defines the number of erroneous cycles the chip tolerates before it goes into the "OUT-OF-LOCK" state

Table 5 Bit assignment of Config register 0.

Wait_time: after start-up and after the transmitter ASIC is properly synchronised, some time ("wait time") must be allowed for the link receiver to synchronise with the incoming data stream. During this period, idle characters are transmitted. The wait time bits (*Config0<4:0>*) allow to control the duration of the wait time (see Table 17, page 39).

Loss_of_lock_time: Since the ASIC will be operating in a radiation environment, it is possible that Single Event Upsets (SEU) will momentarily disturb the internal PLL operation. Internal to the IC, a watchdog circuit monitors the correct operation of the PLL circuit and reinitialises the ASIC operation in the case gross synchronisation errors have occurred. It is however possible that, after an SEU, the ASIC (and the link receiver) can resume normal operation without the intervention of this watchdog circuit. The Loss_of_lock_time bits (*Config0<7:5>*) allow to control the amount of time the error condition must persist before the watchdog circuit will reinitialise the IC (see Table 19, page 40).

Config 1

Bits	Name	Description
<3:0>	PLL_lock_time	Defines the time between the OUT-OF-LOCK state and the LOCKED state.
<4>	en_soft_loss_of_lock	If 1 then the GOL can tolerate a number of erroneous cycles defined by Loss_of_lock time, otherwise the state goes immediately to OUT-OF-LOCK after the first error.
<5>	en_loss_of_lock_count	If 1 then one TX_LOLC cycle is inserted between LOCKED and READY, where the number of loss-of-lock events is transmitted as a 16-bit data word.
<6>	en_force_lock	[Disables lock state machine if 1. Only used for testing/debugging.]
<7>	en_self_test	If 1, a running 16 bit counter generates transmission data

Table 6 Bit assignment of Config register 1.

PLL_lock_time: After power up (or after a synchronisation loss due to an SEU), the lock acquisition state machine requires the PLL lock signal to be stable for a given

amount of time before it considers the PLL to have acquired lock. This elapsed time is controlled by bits *Config1<3:0>* (see Table 18, page 40).

En_soft_loss_of_lock: If the feature "*soft loss of lock*" is disabled (*Config1<4>* = "0") then, whenever a synchronisation loss is detected, a re-initialisation cycle is immediately started. However, if the "*soft loss of lock*" feature is enabled (*Config1<4>* = "1") then a re-initialisation cycle is only started if a number of erroneous cycles, defined by the "*Loss_of_lock_time*", has occurred (see "*Config 0*", Table 5 and Table 19 (page 40)).

En_loss_of_lock_count: the ASIC keeps track of how many synchronisation losses have occurred since the last reset. If the bit *en_loss_of_lock_count* is set to "1" (*Config1<5>*="1"), a special word is transmitted by the ASIC with the "*loss of lock count*" information at the time the chips enters the "READY" state. If the ASIC is operating in the G-Link mode, the "*loss_of_lock_count*" is transmitted as a control word. If the Gbit Ethernet mode is selected, the "*loss_of_lock_count*" is transmitted as an ordinary data word.

En_force_lock: If enabled (*Config1<6>*="1") the IC operates as if the PLL was always locked. This mode is available for testing purposes only.

En_self_test: when this mode is enabled (*Config1<8>* = "1"), the ASIC ignores the data on *din<31:0>*, and generates an internal data sequence. This sequence consists of a cyclic 16-bit count. The transmitted sequence is identical for slow (16 bit) and fast (32 bit) mode: n<7:0>, n<15:8>, n+1<7:0>, n+1<15:8>, n+2<7:0>...

Config 2

Bits	Name	Description
<4:0>	PLL_current	Defines the charge-pump current of the internal phase-locked loop (PLL).
<6:5>	test_sel	[Selects signal to appear on test_analog pad. Only used for testing/debugging.]
<7>	en_flag	Enables flag bits in G-Link mode

Table 7. Bit assignment of Config register 2.

PLL_current: The PLL charge pump current is set at start-up by reading the hardwired ASIC input signals "*conf_i_pll<2:0>*" according to Table 20, page 41. The charge pump current can also be programmed by writing into bits *Config2<4:0>*. For this to have an effect, bit *Config3<7>* has to be set to "1". In this case, the charge-pump current is given by:

$$I_{charge\text{-}pump} = \text{Config2<4:0>} \times 1.25\ \mu A$$

En_flag: when transmitting data in the G-Link mode, a flag bit is always added to each 16 bits of data being transmitted. When the external flag bits are disabled (*Config2<7>* = "0"), the flag bit data is generated internally in the ASIC and alternates between "0" and "1" for every 16-bit word transmitted. If the external flag bits are enabled, then the value of the *flag<1:0>* pins is used. When operating in the 32-bit mode, *flag<1>* is transmitted together with the upper 16 bits (corresponding to *din<31:16>*), and *flag<0>* with the lower 16 bits (*din<15:0>*). In 16-bit mode, only *flag<0>* is used.

Config 3

Bits	Name	Description
<6:0>	LD_current / driver_strength	Defines the bias current for the Laser Driver (LD) or the strength of the 50 Ohm line driver
<7>	use_conf_regs	When 1, the content of the Configuration registers 1 and 2 are used to define the value for PLL_current and LD_current. If 0, the values are derived from the encoded values on the pads.

Table 8. Bit assignment of Config register 3.

LD_current: In laser driver mode (*conf_laser*="1"), and provided that "*use_conf_regs*" = "1", these bits (*Config3<6:0>*) allow programming the laser-diode bias current. The number in bits *Config3<6:0>* translates the laser-diode bias current according to:

$$I_{ld\text{-}bias} = 1\ mA + Config3<6:0> \times 0.4\ mA$$

If "*use_conf_regs*" = "0" then the bias current is set according to the value hardwired in pins *conf_i_pll<2:0>* (see Table 21, page 41).

driver_strength: In the 50 Ω line driver mode (*conf_laser*="0"), and provided that "*use_conf_regs*" = "1", *Config3<6:0>* contains the strength of the line driver, as defined in Table 22, page 42.

Use_conf_regs: When set (*Config3<7> = '1'*) bits *Config3<6:0>* are used to set the laser-diode bias current and bits *Config2<4:0>* are used to set the PLL charge pump current. If this feature is disabled (*Config3<7> = '0'*) then the pins "*conf_i_ld<2:0>*" and "*conf_i_pll<2:0>*" are used to set these currents.

STATUS REGISTERS

Status 0

Bits	Name	Description
<7:0>	loss_of_lock_count	Number of "loss-of-lock" events since last reset.

Table 9. Bit assignment of Status register 0.

Loss_of_lock_count: This register accumulates the number of times the PLL has been detected to be out of lock since the last "RESET". This value can be read from the I2C interface or, if the "*En_loss_of_lock_count*" feature is enabled (see configuration-register "Config 1"), this count will be transmitted each time lock has been regained.

Status 1

Bits	Name	Description
<7:6>	link_control_state_A	Current state of link initialisation logic A
<5:4>	link_control_state_B	Current state of link initialisation logic B

<3:2>	link_control_state_C	Current state of link initialisation logic C
<1>	conf_glink	Reads value on conf_glink pin. If 1, then the chip is configured on G-Link mode, otherwise in Ethernet mode.
<0>	conf_wmode16	Reads value on conf_wmode16 pin. If 1 then the chip accepts 16-bit wide data and transmits with 800 Mbit/s, otherwise 32 bit data and 1.6 Gbit/s speed.

Table 10 Bit assignment of Status register 1.

Link_control_state_A, link_control_state_B, and link_control_state_C: For SEU robustness, the ASIC state machines use triple modular redundancy and majority voting. Bits *Status1<7:6>*, *Status1<5:4>* and *Status1<3:2>* can be read through the I2C or JTAG interface, and they represent the state of link initialisation logic. The meaning of these bits is as follows (see “Initialisation procedure”, on page 18).

00: “OUT-OF-LOCK” state

01: “LOCKED” state

10: “READY” state

11: “TX_LOLC” state

Conf_glink: reports the hardwired value of pin “*conf_glink*”

Conf_wmode16: reports the hardwired value of pin “*conf_wmode16*”

Chapter 3

Register Access via the I2C Bus

The I2C bus protocol defines a standard for an asynchronous serial bus with a maximum transfer rate of one Mbit/s [4].

DATA AND POINTER REGISTER

All data transfer over the I2C bus is performed using only two registers: The **I2C_pointer-register** and the **I2C_data-register**. The *I2C_pointer-register* is 3 bits wide and contains the address of the internal register as defined in Table 4, page 10. When reading the *I2C_data-register*, the content of the register *being addressed by the pointer register* is transferred. Conversely, writing a byte to the *I2C_data-register* in fact writes to the GOL register *addressed by* the *I2C_pointer-register*.

Reading and writing registers

All the registers shown in Table 4 can be accessed over the I2C bus. After a write access, the corresponding configuration register is in general set to the value of the transmitted data byte. A write access to one of the status registers will be ignored.

Each I2C access is performed in two steps:

1) Write the register number in the *I2C_pointer-register*

2) Read or write the *I2C_data-register*

In accordance with the I2C bus specification, each device on the bus is addressed by a 7-bit wide I2C device address. Each GOL chip occupies two consecutive positions in the 7-bit I2C address space. Hence, it is possible to address a total of 64 devices in one system. The 7 Bit I2C address is derived from the content of the value on the *i2c_addr<6:1>* pins in the following way:

I2C access register name	resulting 7 bit I2C address
I2C_pointer	{i2c_addr<6:1> , 0 }
I2C_data	{i2c_addr<6:1> , 1 }

Table 11 I2C address calculation.

Example of a read operation

Assuming, that we intend to read the content of the ***Status 0*** register, and the *i2c_addr<6:1>* pins are hard-wired to "110001". The procedure is as follows:

First, by writing the value 4 (=register number according to Table 4, page 10) on I2C address 1100010, the pointer register is set. This is then followed by a read access onto the I2C address 1100011 (the data register), which delivers the content of the *Status 0* register.

Chapter 4

Register Access via the JTAG Bus

The JTAG standard defines a serial communication protocol for testing and programming purposes.

JTAG INTERFACE FUNCTIONALITY

In the GOL chip, the JTAG interface supports three tasks

- Boundary scan
- Access to internal configuration and status registers
- Test of internal logic

The different functions are reflected in a number of scan registers, which are listed in Table 12. A specific scan path can be selected by writing its 5-bit code into the instruction register (IR) inside the on-chip JTAG controller [5]. Table 12 lists all available scan registers: The EXTEST register is used for boundary scan testing, which is described in Detail in Chapter 9, "JTAG boundary scan". The DEVICE_ID register can be used to read the JTAG identification number. The CONF_RW and CONF_R registers are used to transfer data from (and to) the configuration and status registers. The CORETEST register provides a simple means of verifying the correct function of the internal logic for production testing. The BYPASS register is used to short-circuit the scan-path if, for example, JTAG testing is used with more devices in a system.

Scan registers

JTAG scan register	JTAG Instr. reg. content	Length of scan path
EXTEST	00000	60
DEVICE_ID ("$B416E6B3")	00001	32
CONF_RW	01001	55
CONF_R	01010	55
CORETEST	01011	432
BYPASS	11111	1

Table 12. The available JTAG registers in the GOL chip together with the required instruction register code.

Reading and Writing the configuration and status registers

Two scan registers are defined for accessing the configuration and status registers, CONF_RW and CONF_R. While the first one is for read and write access, the latter only reads the data from the configuration and status registers. The only difference in the implementation of these two is that in the case of CONF_R the update cycle ("Update-DR", see [5]) does not lead to a change in the configuration registers. Using the CONF_RW scan path, the current register content is shifted out at the same time

that a new register content is shifted in the scan path. During the Update-DR cycle, the configuration registers are then loaded with the new values.

Register scan path definition

The shifting order for the CONF_RW and CONF_R scan paths is defined in Table 13. Besides the two status registers and the four configuration registers, the scan path contains a seven bit hamming check-sum of the 32 bits configuration data. For write access, only the values of the configuration registers are significant, values written to the status registers or the hamming check-sum are ignored.

Position shift OUT	Position shift IN	Signal/Register content
0	54	status 0<0>
1	53	status 0<1>
...	...	...
7	47	status 0<7>
8	46	status 1<0>
...	...	...
15	39	status 1<7>
16	38	config 0<0>
...	...	...
23	31	config 0<7>
24	30	config 1<0>
...	...	...
31	23	config 1<7>
32	22	config 2<0>
...	...	...
39	15	config 2<7>
40	14	config 3<0>
...	....	...
47	7	config 3<7>
48	6	hamming<0>
...	...	...
54	0	hamming<6>

Table 13. Scan path definition for register access using the CONF_RW and CONF_R modes.

Chapter 5

ASIC Operation

During normal operation, the GOL chip reads a 32 (or 16 bit) word from the *din<31:0>* bus, and puts out its corresponding serial bit stream on either the laser or the 50 Ω line driver. An internal phase-locked loop (PLL) generates the high-speed clocks from the external clock source, which is the 40.08 MHz LHC clock. The transmission is therefore completely synchronous with fixed latency. If the PLL is not properly locked to the clock signal, then the data values on *din<31:0>* are ignored, and a proper idle symbol is transmitted. An initialisation procedure is performed, which is described below

DATA INTERFACE

The data interface samples the following input signals

- *din<31:0>*
- *flag<1:0>*
- *dav / tx_en*
- *cav / tx_er*
- *force_ff0.*

If in 32 bit mode, the data interface also converts 32 bit wide data into 16 bit words, which are then fed to either the G-Link or the 8B/10B encoder. The signals have to fulfil timing (setup and hold time) requirements with respect to either the rising or the falling edge of *clkLHC* (See Chapter 6, "Signals and Timing", Figure 4 and Figure 5). The *conf_negedge* pin specifies if the positive or negative edge of *clkLHC* is to be used to sample the input signals.

INITIALISATION PROCEDURE

A state machine controls the lock acquisition behaviour of the circuit. Its state diagram is depicted in Figure 2, and the different states are listed in Table 14. After a reset or after the ASIC's internal PLL has lost lock, the lock monitoring state machine enters the "OUT-OF-LOCK" state.

PLL lock time

Having entered the "OUT-OF-LOCK" state, the state machine waits until it is certain that the phase-locked loop has acquired phase lock with the reference clock. This is done by counting "m" consecutive cycles in which the PLL asserts the "instant_lock" internal flag. (The "instant_lock" signal originates in the analogue circuitry of the PLL. Although a "1" on this signal is not necessarily an indicator of a properly locked PLL, a "0" is a certain indicator that the PLL is not locked.) The count "m" is defined by the bits "*PLL_lock_time*" (see "Config 0", page 11 and Table 18, page 40). After "m"

cycles, the state machine enters the "LOCKED" state. During the "OUT-OF-LOCK" and “LOCKED” states, an “IDLE” symbol[2] is transmitted.

Wait time

The chip stays in the "LOCKED" state for a period of time long enough to assure that the receiver can acquire phase lock with the incoming bit stream.

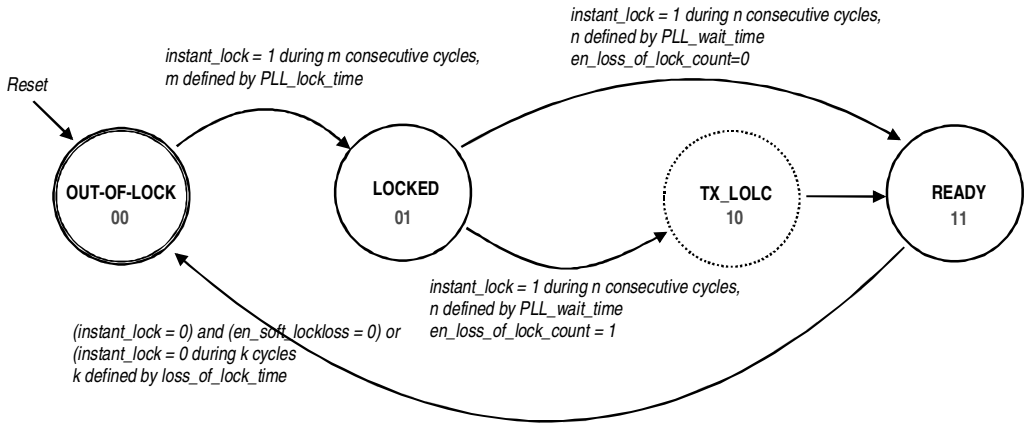

Figure 2 Lock acquisition and monitoring state diagram

The duration of this period, "n" cycles, is controlled by the *Wait-time.* The wait time is derived from the value in Config 0, as defined in Table 17 (page 39). After this time has elapsed (and provided that the "instant_lock" signal remains "1") the transmitter enters the "READY” state were transmission of data is resumed.

Loss-of-lock count feature

The "READY" state is entered directly from the “LOCKED” state if the “*Transmit Loss of Lock Count*” feature is disabled, that is, *en_loss_of_lock_count* = “0” (see Config 1 (page 11)). If, however, this feature is enabled (*en_loss_of_lock_count* = “1”) then, before entering the “READY” state, an intermediary “TX_LOLC” cycle is inserted. In this state, the number of times a loss of synchronisation has been detected since the last “RESET” is transmitted over the link (see Config 1). In G-Link mode this is a 16 bit control word, in Ethernet mode the loss-of-lock counter is transmitted as an ordinary data word.

Control of Loss-of-lock behaviour

The ASIC has been designed to operate in a radiation environment, where Single Event Upsets (SEU) will eventually disturb the internal PLL operation. To decide if the PLL is operating properly the “*instant_lock*” signal is constantly monitored. During normal operation this signal should have the logical value “1”. However, when phase lock is lost this signal takes the logic value “0”. The user has the choice between a hard (*en_soft_loss_of_lock* = “0”) or a soft (*en_soft_loss_of_lock* = “1”) detection of synchronisation loss (see Config 1, page 11). If the hard decision is used and *instant_lock* = “0” for a single cycle, the lock monitoring state machine enters the “OUT-OF-LOCK” state. If the *soft* decision mode is selected, the lock monitor state machine has to detect that the signal *instant_lock* is “0” for "k" cycles (consecutive or not) before it goes into the "OUT-OF-LOCK" state. This count is reset to zero if 1024

[2] The actual value of the idle sequence depends on the operation mode (G-Link or Ethernet).

consecutive cycles with instant_lock = “1” are detected before the count reaches the value “k”. The number "k” corresponds to the "*loss_of_lock_time*" in *configuration register 0* (see page 11). The mapping between the value of *loss_of_lock_time* and *k* is defined in Table 19 on page 40.

State definition table

State	link control state	Description	transmitted symbol	ready pin
OUT-OF-LOCK	00	PLL is unlocked	IDLE	0
LOCKED	01	PLL has acquired lock	IDLE	0
TX_LOLC	11	Transmit content of loss_of_lock_counter	loss_of_lock_count	0
READY	10	Normal operation	Data from the data bus	1

Table 14 Table of states of the lock acquisition state machines.

Chapter 6

Signals and Timing

This chapter describes the GOL signal pins and the most important timing relations among these signals.

LIST OF SIGNALS

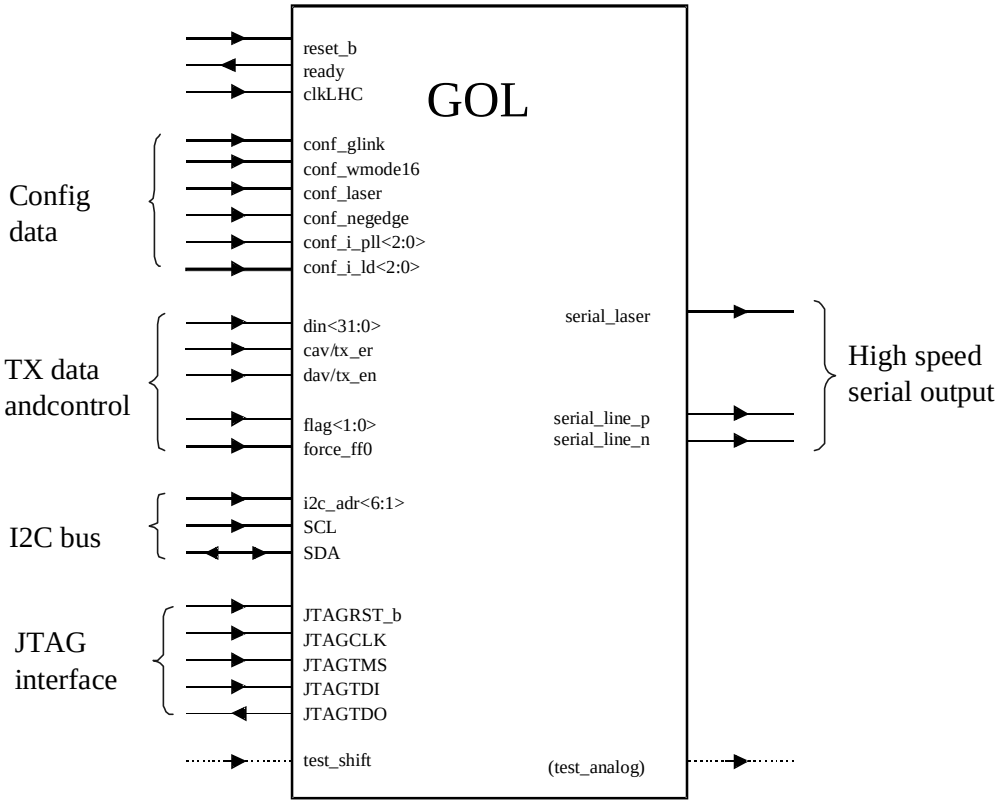

Figure 3 ASIC external signals overview.

The various signals of the GOL chip are displayed in Figure 3. The signals can be divided into several groups:

- Configuration data pins
- Reference clock
- Transmit data and transmit control signals
- I2C interface
- JTAG interface
- High speed serial output signals
- Test

reset_b

Active “0” master reset signal.

ready

This signal indicates that the chip is in the READY state. In this state, the data available on the input data bus “*din<31:0>*” is transmitted.

clkLHC

40.08 MHz LHC reference clock.

conf_glink

Selects G-Link (CIMT encoding) mode when “1”, or Ethernet (8B/10B encoding) mode when “0”.

conf_wmode16

Selects slow (16 bit input, 800 Mbit/s) mode when “1”, or fast (32 bit input, 1.6 Gbit/s) mode when “0”.

conf_laser

Selects serial laser driver output when “1”, or differential 50 Ω line driver when “0”.

conf_negedge

Selects clock edge to validate input data on *din<31:0>* bus. If “1”, *din* is validated on the falling edge of *clkLHC*, otherwise on the rising edge. See Figure 4 and Figure 5.

conf_i_pll<2:0>

Selects the phase-locked loop (PLL) charge pump current, provided that the *use_conf_regs* bit in the *Config3* register is set to “0”. Otherwise, the value is directly taken from the *PLL_current* bits of the configuration-register “*Config2*”.

conf_i_ld<2:0>

Selects the laser driver bias current, if the *use_conf_regs* bit in the Config3 register is set to “0”. Otherwise, the value is directly taken from the *LD_current* bits of the configuration register “*Config1*”.

din<31:0>

Input data bus. In the fast (= 32 bit) mode, all 32 bits are used. In the slow (= 16 bit) mode, only bits *din<15:0>* are used.

cav / tx_er

In G-Link mode, *cav* ("control available") indicates the availability of a control data word. In Ethernet mode, *tx_er* together with *tx_en* determines the type of data to be sent according to Table 3.

dav / tx_en

In G-Link mode, *dav* ("data available") indicates the availability of a data word. In Ethernet mode, *tx_en* together with *tx_er* determines the type of data to be sent according to Table 3.

flag<1:0>

A flag bit can be sent as additional data in one 20 bit G-Link frame, hence the number of transmitted data bits per 20-bit frame becomes seventeen. *Flag<0>* is transmitted together with the lower 16 data bits (*din<15:0>)*, while *flag<1>* is sent with the upper 16 data bits (*din<31:16>)*. *flag<1>* is only used in the fast (=32 bit) mode. The *flag<1:0>* pins are ignored in the Ethernet mode.

force_ff0

Forces the transmission of the ff0 symbol while in G-Link mode. This signal is ignored in the Ethernet mode.

i2c_addr<6:1>

The six most significant bits of the 7 bit I2C address.

SCL

I2C clock signal.

SDA

Bi-directional I2C data signal. This open-drain signal must be connected to a pull-up resistor.

JTAGTCK

JTAG test clock.

JTAGTDI

JTAG test data in.

JTAGTDO

JTAG test data out.

JTAGTMS

JTAG test mode select.

JTAGTRST_b

JTAG test reset. Unless a JTAG controller is present, this pin should be connected to ground.

ld_cathode

High-speed (0.8 or 1.6 Gbit/s) serial output, which is active when *conf_laser* = “1”. This pin should be directly connected to the laser diode. In order to reduce inductance, the pin is duplicated.

ld_anode

Laser diode anode connection.

serial_line_p, serial_line_n

High-speed (0.8 or 1.6 Gbit/s) differential, serial output, which is active when the line driver output is selected (*conf_laser* = “0”).

test_shift

Selects test shift mode. This pin is only used for prototype testing, and must be connected to ground.

RECOMMENDED OPERATING CONDITIONS

		MIN	TYP	MAX	UNIT
V_{DD}	Supply voltage	2.25	2.5	2.75	V
V_{IH}	High-level input voltage	V_{DD}-0.5			V
V_{IL}	Low-level input voltage			0.5	V
T_A	Operation free-air temperature	-10	27	75	°C

Table 15 Recommended operating conditions.

TIMING CHARACTERISTICS

		MIN	TYP	MAX	UNIT
t_{clkLHC}	clkLHC period		24.95		ns
d_{clkLHC}	clkLHC duty cycle	40		60	%
jpp_{clkLHC}	Allowable peak-peak jitter on clkLHC			100	ps
$t_{setup,p}$[3]	Data[4] stable to clkLHC rising edge	5			ns
$t_{setup,n}$[5]	Data stable to clkLHC falling edge	5			ns
$t_{hold,p}$	Data hold time w.r.t. clkLHC rising edge	3			ns
$t_{hold,n}$	Data hold time w.r.t. clkLHC rising edge	3			ns
$t_{lat(32),1}$	clkLHC until first byte transmitted, fast (32 bit) mode[6]	35	40	45	ns
$t_{hat(32),4}$	clkLHC until fourth byte transmitted, fast (32 bit) mode[7]	54	59	64	ns
$t_{lat(16),1}$	clkLHC until first byte transmitted, slow (16 bit) mode	68	73	78	ns
$t_{lat(16),2}$	clkLHC until second byte transmitted, slow (16 bit) mode	80	85	90	ns
$t_{r,I2C}$	Rise time on SCL, SDA			30	ns
$t_{r,I2C}$	Fall time on SCL, SDA			30	ns

Table 16 Timing characteristics.

[3] Applies if rising edge of clkLHC is selected to validate data (conf_negedge = 0). See Figure 4.
[4] Data signals are: din<31:0>, dav/tx_en, cav/tx_er, flag<1:0>, force_ff0. See Figure 5
[5] Applies if falling edge of clkLHC is selected to validate data (conf_negedge = 1).
[6] See Figure 6.
[7] See Figure 7.

Data interface timing

The data interface timing defines setup and hold times with respect to *clkLHC* for pins *din<31:0>*, *flag<1:0>*, *dav/tx_en*, *cav/tx_er* and *force_ff0* . The *conf_negedge* pin defines if the rising or falling edge of *clkLHC* is used to validate the data.

Transmit latency

The transmit latency time is in principle defined as the time from the (rising or falling[8]) clock edge of *clkLHC* to the time when the first byte is completely serialised. A second latency time is also given, defined as the time from *clkLHC* until complete serialisation of the last byte.

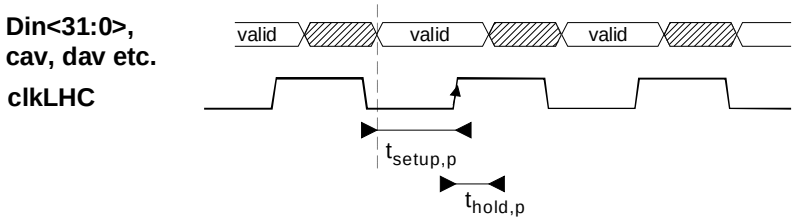

Figure 4 Definition of data setup and hold times for the case that the rising edge of *clkLHC* is used to validate the data (*conf_negedge* = 0).

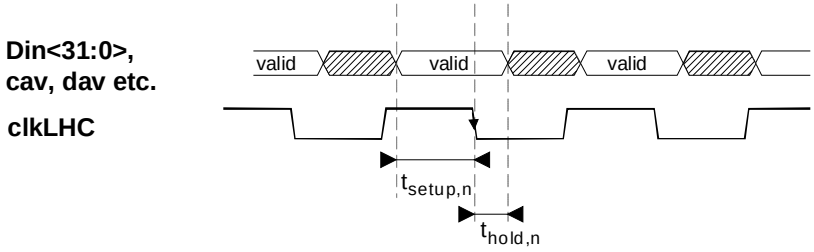

Figure 5 Definition of data setup and hold times for the case that the falling edge of *clkLHC* is used to validate the data (*conf_negedge* = 1).

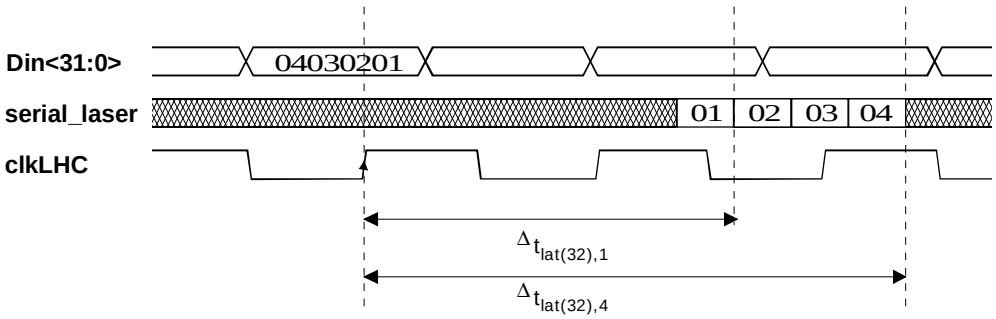

Figure 6 Definition of latency times for the fast (=32 bit) mode.

[8] Depending on *conf_negedge*.

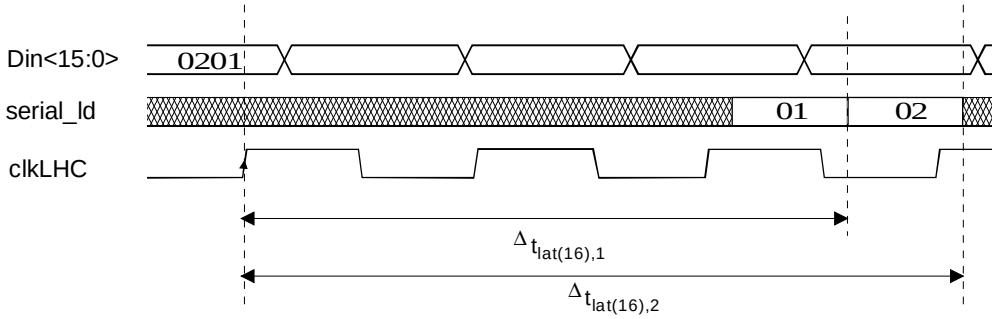

Figure 7 Definition of latency times for the slow (=16 bit) mode.

Chapter 7

Measures against Radiation effects

Owing to the fact that the ASIC will be operating in a highly radioactive environment, two types of phenomena are of concern. The first type are accumulated dose effects, that can, in the long run, permanently degrade the performance of the IC, or cause the device to fail, the other type are single event effects.

ACCUMULATED DOSE EFFECTS

The utilisation of a deep sub-micron CMOS technology in conjunction with special layout techniques, such as gate-all-around transistors and carefully placed guard-rings, proved to be an effective measure to counteract cumulative effects. These techniques are extensively described in references [6] and [9].

SINGLE EVENT UPSETS (SEU)

Momentary perturbations on the chip's internal signals due to radiation are called Single Event Upsets (SEU). In digital circuits, this can result in a flipped bit in a register, or a glitch in the logic, which eventually can be latched in a register. The IC thus either displays erroneous data at its outputs, or can be stuck in a wrong state, which would then require the control system to issue a reset or to reprogram the circuit.

Concerning single event effects, digital signals can be classified into three groups: configuration data, state vectors and pipelined data. While configuration data usually changes very rarely (or not at all) during the operation of the circuit, state vectors and pipelined data are updated at every clock cycle. The difference between state vectors and pipelined data is that, in the former case, a change due to an error also affects subsequent cycles, while an error on pipelined data only changes the output during a single cycle, thus leaving the state of the circuit intact.

When aiming for robust operation it is obvious that protection against single event effects is mandatory for configuration data and state vectors, since an error in these signals changes the operation of the circuit on a long-term perspective. In the ASIC, several different approaches were used to minimise the impact of SEU on the circuit operation:

Hard-wired configuration data

First, since hard-wired logic values cannot be flipped and – apart from short-term glitches – are guaranteed to stay constant, all configuration bits, which are expected not to change during the operation of the circuit, are connected to external pins. This concerns those configuration bits that decide if the circuit operates in G-Link or 8B/10B mode (conf_glink), the selection of either the laser or the line driver (conf_laser), and the choice of the bit rate (conf_wmode16).

Hamming-protected configuration register

Second, all other configuration settings (i.e. those which are expected to change during the lifetime of the experiment) are stored in a Hamming code protected memory [10]. This e.g. concerns the laser pre-bias current that might have to be adjusted to compensate for laser ageing. Every time a Hamming code violation is detected, a correction state machine is activated and the register contents restored. This should result in an error condition that lasts for less than a single LHC master clock cycle. Previous experience, where a similar implementation was used in a different ASIC [11] has proven this method to be robust.

Triple modular redundancy

Third, protecting the data path and the IC control logic requires error correction to be made at clock pace, i.e. at 40 MHz or at 80 MHz, depending on the configured transmission rate. For these functions, it was decided to use triple modular redundancy and a voting scheme according to the general concept illustrated in Figure 8.

In this scheme, every logic function is implemented three times, and the state vectors of the three identical state-machines are daisy-chained. Thus, the Finite State Machines (FSM) marked (1) and (2) in the figure, calculate not their own next state vector but those of their neighbouring circuit. Finally, the next-state vector that is fed to the FSM (1) is calculated by a majority voter circuit that uses as inputs the next-state vectors calculated by the three FSM. If the error rate is not excessively large, only a single error should be present for the duration of three clock cycles. From this circuit configuration, it results that the error condition will not survive in the system for more than a three cycles and, consequently, the state sequence will not be disturbed by single errors. One particular advantage of the depicted scheme is that the state machine restores its correct state also in the case of a single event upset in the voting circuit itself. The scheme was developed as an alternative to a structure that would require the triplication of all the voting devices, demanding additional logic and wiring complexity.

Particular care has to be taken for the clock signal feeding the finite state machines. If the same clock signal is used to drive the three state machines then a glitch in this signal can cause all FSM's to be wrong, thus creating an unrecoverable error condition. However, due to the typical high loading of the clock lines, glitches in these signals are very unlikely. Nonetheless, if the loading of the clock lines is low, the scheme shown in Figure 8, where each FSM receives the clock signal from its own clock buffer, can be used. In this case, an upset in only one of the clock buffers will not disturb the operation of the other circuits.

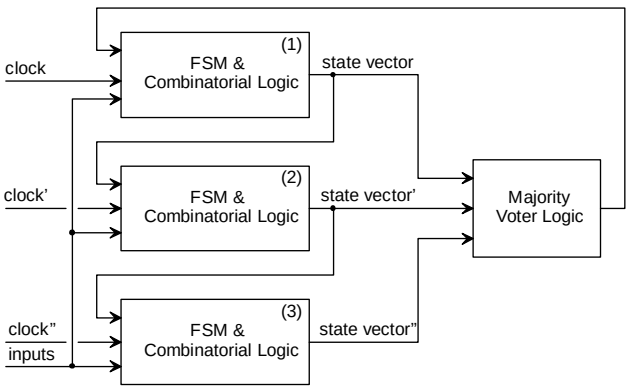

Figure 8 Triple redundancy voting scheme

Up-sized analog components

Fourth, the analogue components (PLL, Laser and Line Drivers) and high-speed digital blocks (Serializer and Clock generator) of the ASIC are not natural candidates to triple modular redundancy. That is either because their operation is intrinsically analogue, or because their fast operation is not compatible with the extra delay penalty introduced by the voting mechanism.

It has been shown [12] that by up-sizing the MOS transistor widths it is possible to increase the Linear Energy Transfer (LET) threshold thus reducing the error rate for a given radioactive environment. This technique was used in this IC to improve the resistance to SEU's of the Serializer and Clock-Generators circuits. This size increase was made relative to an existing prototype that operates at 1.2 Gbit/s [13].

The Line and Laser Driver analogue cells already run at sufficiently high current levels, hence their sensitivity to ionising particles is very low. However, the PLL circuit involves cells with low current levels and the circuit had to be adapted to increase its robustness against SEU.

Chapter 8

Packaging and Pin Assignments

The GOL ASIC package is still undefined.

PIN ASSIGNMENTS

The following two tables list the IC pin names and numbers sorted by the bonding pad number on the chip.

Pin assignments: sorted by pin number

IC B. Pad #	PIN #	Name	Type	Description
1		VDDA1	pwr	
2		VDDA1	pwr	
3		GND	pwr	
4		GND	pwr	
5		ld_cathode		laser driver output
6		ld_cathode		laser driver output
7		ld_anode	pwr	
8		ld_anode	pwr	
9		VDDA1	pwr	
10		VDDA1	pwr	
11		GND	pwr	
12		GND	out	
13		serial_line_n	out	differential bit-stream output
14		serial_line_n	out	differential bit-stream output
15		serial_line_p	out	differential bit-stream output
16		serial_line_p	out	differential bit-stream output
17		VDDA	pwr	
18		VDDA	pwr	
19		GND	pwr	
20		GND	pwr	
21		i2c_addr<1>	in	I2C device address
22		i2c_addr<2>	in	I2C device address
23		i2c_addr<3>	in	I2C device address
24		i2c_addr<4>	in	I2C device address
25		i2c_addr<5>	in	I2C device address
26		i2c_addr<6>	in	I2C device address
27		GND	pwr	
28		conf_glink	in	“1”: G-Link; “0”: 8B/10
29		conf_laser	in	“1”: laser-driver; “0”: line-driver
30		VDDA2	pwr	PLL power
31		GND	in	
32		conf_wmode16	in	“1”: 16-bit (0.8 Gbit/s); “0”: 32-bit (1.6 Gbit/s)
33		conf_i_pll<0>	in	selects PLL bias current

34		conf_i_pll<1>	in	"
35		conf_i_pll<2>	in	"
36		GND	pwr	
37		VDD	pwr	
38		conf_i_ld<0>	in	selects laser driver bias current
39		conf_i_ld<1>	in	"
40		conf_i_ld<2>	out	"
41		test_shift	in	enables test of internal logic
42		JTAGTCK	in	JTAG clk
43		JTAGTMS	out	JTAG mode select
44		JTAGTDO	in	JTAG data out
45		JTAGTDI	in	JTAG data in
46		JTAGTRST_b	in	JTAG reset
47		GND	pwr	
48		VDD	pwr	
49		test_analog	in	test output
50		ready	in	transmitter ready
51		SDA	in / out	I2C data
52		SCL	in	I2C clock
53		GND	pwr	input data
54		VDD	pwr	
55		din<0>	in	input data<15:0>
56		din<1>	in	"
57		din<2>	in	"
58		din<3>	in	"
59		din<4>	in	"
60		din<5>	in	"
61		din<6>	in	"
62		din<7>	in	"
63		din<8>	in	"
64		din<9>	in	"
65		din<10>	in	"
66		din<11>	in	"
67		din<12>	in	"
68		din<13>	in	"
69		din<14>	in	"
70		din<15>	in	"
71		din<16>	in	input data
72		din<17>	in	"
73		din<18>	in	"
74		din<19>	in	"
75		din<20>	in	"
76		din<21>	in	"
77		din<22>	in	"
78		din<23>	in	"
79		GND	pwr	
80		VDD	pwr	
81		din<24>	in	input data
82		din<25>	in	"
83		din<26>	in	"
84		din<27>	in	"
85		GND	pwr	"
86		VDD	pwr	"
87		din<28>	in	"
88		din<29>	in	"
89		din<30>	in	"

90		din<31>	in	"
91		flag<0>	in	flag bit<0> (G-Link mode only)
92		flag<1>	in	flag bit<1> (G-Link mode only)
93		cav	in	control data available
94		dav	in	data available
95		force_ff0	in	force ff0 symbol (G-Link mode only)
96		VDDA2	pwr	PLL power
97		GND	pwr	
98		conf_negedge	in	Data is read on falling edge of clk when “1”. On the rising edge if “0”
99		reset_b	in	master reset
100		clkLHC	in	40.08 MHz input clock

Chapter 9

JTAG Boundary-Scan

The GOL chip implements a subset of the JTAG/IEEE 1149.1 standard (see for instance [5]) providing the capability for board-level connectivity tests. A list of the implemented JTAG scan registers can be found in Table 12 (page 16).

JTAG Device ID

The JTAG logic includes a Device Identification Register and the device identification number is:

ID = “B416E6B3” (HEX)

Boundary Scan Register

The Boundary Scan Register (BSR) includes all the I/O signals with exception of the analogue (laser and line driver) outputs and the clkLHC signal.

Boundary scan register read out order

Order for Shift OUT	PIN #	Name	Type	Description
0		sda_out	out	I2C data (SDA) output
1		ready	out	GOL in ready state
2		test_shift[9]	in	direct shift test active
3		sda_in	in	I2C data (SDA) input
4		scl_in	in	I2C clock (SCL) input
5		i2c_addr<1>	in	I2C base address
6		i2c_addr<2>	in	"
7		i2c_addr<3>	in	"
8		i2c_addr<4>	in	"
9		i2c_addr<5>	in	"
10		i2c_addr<6>	in	"
11		force_ff0	in	forces ff0 symbol (only used in G-Link mode)
12		flag_i<0>	in	flag bits (only used in G-Link mode)
13		flag_i<1>	in	"
14		cav / tx_er	in	control available / transmit error
15		dav / tx_en	in	data available / transmit enable
16		din<0>	in	input data bus
17		din<1>	in	input data bus
18		din<2>	in	"
19		din<3>	in	"
20		din<4>	in	"
21		din<5>	in	"
22		din<6>	in	"
23		din<7>	in	"
24		din<8>	in	"

[9] Pin test_shift should always be connected to ground, because it can disturb the correct operation of the JTAG logic.

25		din<9>	in	"
26		din<10>	in	input data bus
27		din<11>	in	"
28		din<12>	in	"
29		din<13>	in	"
30		din<14>	in	"
31		din<15>	in	"
32		din<16>	in	"
33		din<17>	in	"
34		din<18>	in	"
35		din<19>	in	"
36		din<20>	in	"
37		din<21>	in	"
38		din<22>	in	"
39		din<23>	in	"
40		din<24>	in	"
41		din<25>	in	"
42		din<26>	in	"
43		din<27>	in	"
44		din<28>	in	"
45		din<29>	in	"
46		din<30>	in	"
47		din<31>	in	"
48		conf_i_pll<0>	in	PLL current selection
49		conf_i_pll<1>	in	"
50		conf_i_pll<2>	in	"
51		conf_i_ld<0>	in	Laser driver current selection
52		conf_i_ld<1>	in	"
53		conf_i_ld<2>	in	"
54		conf_negedge	in	Selects falling clock edge to sample din
55		conf_wmode16	in	Selects 16 bit (=slow) or 32 bit (=fast) mode
56		conf_laser	in	Selects laser (1) or line driver (0)
57		conf_glink	in	Selects G-Link (1) or Ethernet (0) mode
58		reserved	-	set to 0 when read
59		reset_b	in	Master reset

Chapter 10

Verilog evaluation model

A Verilog model of the GOL chip, which contains the full functionality of the circuit, can be used for system simulation purposes. In addition, a test environment provides test stimuli for the chip and allows executing commands, for example, writing register values via the I2C bus.

The virtual test environment

A block diagram of this virtual test environment is shown in Figure 9. It contains the following modules:

- **gol** : The GOL chip
- **jtag_controller** : generates the signals for the JTAG bus
- **data_gen** : generates clkLHC and input data on din<31:0>. The generated pattern on din<31:0> is a simple sequence of four byte counters: $03020100, $07060504, etc.
- **i2cc** : I2c bus controller
- **fc_receiver** : receives the serialized data from the ld_cathode pin and decodes the 8B/10B code.

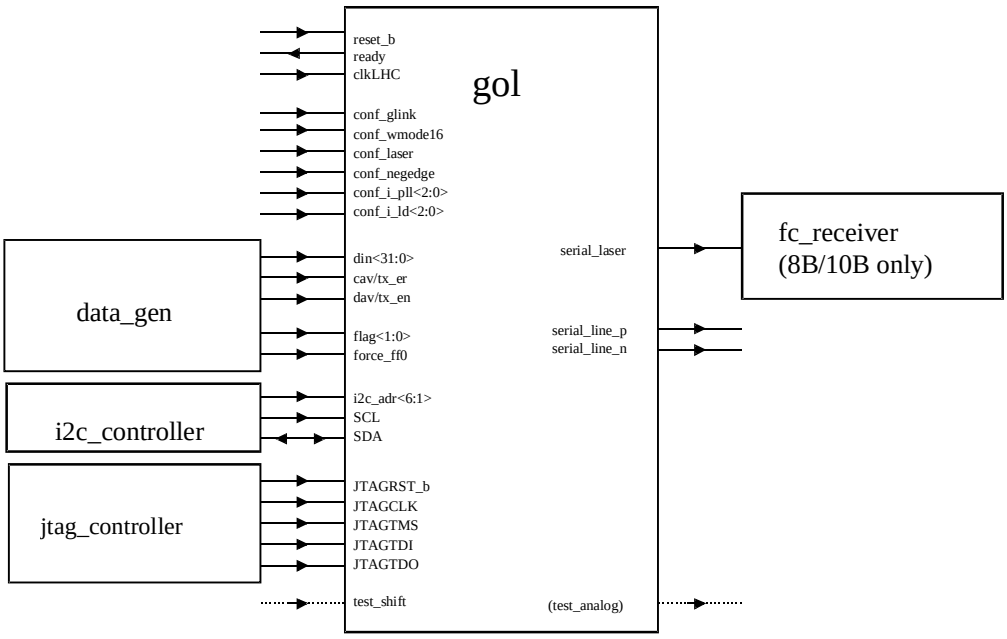

Figure 9 The Verilog virtual test environment.

Verilog files

gol.vp : this file contains a Verilog model of the entire GOL chip

gol_eval.v : contains the virtual test environment (all modules except the GOL chip).

To start the simulation platform using Verilog-XL's graphical user interface type

```
verilog gol_eval.v +gui
```

Then, inside the test fixture, pause the simulation, type

```
test.help;
```

and continue the simulation. You will then get an overview of the available commands. Since the commands are implemented as Verilog tasks they can only be typed in when the simulation is paused. After command entry, the simulation has to be resumed. In general, the commands have to be prefixed with "**test."** to set the right scope for their execution.

Commands

help;

Displays a list of the available commands.

do_reset;

Creates a reset pulse on the RESET_B pin.

i2c_read_reg (I2C_DEV_ADDR, REG_NR);

Reads register content **REG_NR** from chip with I2C address **I2C_DEV_ADDR**.

i2c_read_regs (I2C_DEV_ADDR);

Reads the register file (4 config and 2 status register bytes) from chip with I2C address **I2C_DEV_ADDR**, and calls display_regs to display them.

i2c_write_reg (I2C_DEV_ADDR, REG_NR, DATA);

Writes **DATA** onto register **REG_NR** in chip with I2C address **I2C_DEV_ADDR.**

display_regs;

Decodes and displays the data from the register file.

jtag_read_regs;

Reads the register file (4 config and 2 status register bytes) via the JTAG interface, and calls display_regs to display them.

jtag_write_regs (CONF_REG0, CONF_REG1, CONF_REG2, CONF_REG3);

Writes configuration registers 0-3 via the JTAG interface.

jtag_boundary_scan_test;

Tests the JTAG boundary scan function.

fc_receiver_start;

Starts to display the received Fibre-channel (=Ethernet) data bytes. Data characters have the form <Dx.y>, control characters <Kx,y>.

`fc_receiver_stop;`

Stops displaying received data.

Remarks

Changing the value on configuration pins

The signals going to and from the GOL are defined in the test environment (scope "test") in capital letters (e.g. RESET_B). You can change the value on configuration pins by assigning the required value to the provided register. For example, you can stop the simulation and type

```
test.CONF_GLINK = 1;
```

in order to switch to G-Link mode.

Reducing the wait time until READY

Since the initial settings for the wait time is 13, the GOL checks the correct lock of its internal PLL lock for an impractically large time in the simulation. This can be circumvented by writing a smaller number via the I2C or the JTAG bus.

For example, you can type

```
test.i2c_write_reg (0, 0, 33);
```

to change the wait_time setting to 1. The chip will then go shortly afterwards into the READY state.

Appendix A

Timing and Currents Tables

Wait time encoding table

Wait time	# Cycles	time (fast mode)	time (slow mode)
0	2	12.5 ns	25 ns
1	3	37.5 ns	75 ns
2	5	62.5 ns	125 ns
3	9	112.5 ns	225 ns
4	17	212.5 ns	425 ns
5	33	412.5 ns	825 ns
6	65	812.5 ns	1.6 us
7	129	1.6 us	3.2 us
8	257	3.2 us	6.4 us
9	513	6.4 us	12.8 us
10	1,025	12.8 us	25.6 us
11	2,049	25.6 us	51.2 us
12	4,097	51.2 us	102.4 us
13	8,193	102.4 us	204.8 us
14	16,385	204.8 us	409.6 us
15	32,769	409.6 us	819.2 us
16	65,537	819.2 us	1.64 ms
17	131,073	1.64 ms	3.28 ms
18	262,145	3.28 ms	6.55 ms
19	524,289	6.55 ms	13.12 ms
20-31	1,048,577	13.12 ms	26.24 ms

Table 17 Wait time encoding.

PLL_lock_time encoding table

Wait time	# Cycles	time (fast mode)	time (slow mode)
0	1	12.5 ns	25 ns
1	2	37.5 ns	75 ns
2	4	62.5 ns	125 ns
3	8	112.5 ns	225 ns
4	16	212.5 ns	425 ns
5	32	412.5 ns	825 ns
6	64	812.5 ns	1.6 us
7	128	1.6 us	3.2 us
8	256	3.2 us	6.4 us
9	512	6.4 us	12.8 us
10-15	1,024	12.8 us	25.6 us

Table 18 PLL lock time encoding.

Loss_of_lock_time encoding table

Loss_of_lock time setting	# Cycles	time (fast mode)	time (slow mode)
0	0	0 ns	0 ns
1	2	25 ns	59 ns
2	4	50 ns	100 ns
3	8	100 ns	200 ns
4	16	200 ns	400 ns
5	32	400 ns	800 ns

Table 19 Loss_of_lock time encoding.

Charge-pump current encoding table

Hard-wired pin settings conf_i_pll<2:0>	Equivalent Configuration-register 2 setting Config2<4:0>	Charge-pump current [μA]
000	00001	1.25
001	00010	2.5
010	00100	5
011	01000	10
100	10001	21.25
101	10010	22.5
110	10100	25
111	11000	30

Table 20 Charge-pump current encoding.

Table 20 shows the charge-pump currents that can be obtained by hardwiring the *conf_i_pll<2:0>* pins. Other values of currents are possible by writing to the configuration register 2. In this case, the charge-pump current will be given by:

$$I_{\text{charge-pump}} = \text{Config2<4:0>} \times 1.25\ \mu A.$$

Laser-diode bias current

Hard-wired pin settings conf_i_ld<2:0>	Equivalent Configuration-register setting Config3<6:0>	Laser-diode bias current [mA]
000	0000001	1.4
001	0000010	1.8
010	0000100	2.6
011	0001000	4.2
100	0010000	7.4
101	0100000	13.8
110	1000000	26.6
111	1111111	31.8

Table 21. Laser-diode bias current.

Table 21 shows the laser-diode bias currents that can be obtained by hard-wiring the pins *conf_i_ld<2:0>*. Other values of bias currents are possible by writing to the configuration-register 3. In this case, the bias current will be given by:

$$I_{\text{ld-bias}} = 1\ \text{mA} + \text{Config3<6:0>} \times 0.4\ \text{mA}.$$

Line-driver strength selection

The strength of the 50 Ω line driver is selected by the same register (*Config3<6:0>*) and pins (*conf_i_ld<2:0>*) as used for the selection of laser diode bias current. This is possible because only one (either laser or line driver) can be active at a given moment. If the configuration register is selected, then the driving strength is proportional to the number of ones in the binary value *Config<6:3>.* Otherwise, if the *conf_i_ld<2:0>* pins are used (*use_conf_regs*="0", see *Config* 3), the strength is derived from Table 22:

Pin settings conf_i_ld<2:0>	Configuration-register 3 Config3<6:0>	Line driver strength
000 / 001 / 010	0000 xx	0
011	0001 xx	1
100	0010 xx	1
101	0100 xx	1
110	1000 xx	1
	0011 xx	2
	0101 xx	2
	0110 xx	2
	1001 xx	2
	1010 xx	2
	1100 xx	2
	0111 xx	3
	1011 xx	3
	1101 xx	3
	1110 xx	3
111	1111 xx	4

Table 22 Line driver strength encoding.

Appendix B

Configuration and Status register summary

Register overview

I2C reg. address	Register name	Default content (after reset)
Configuration registers		
0	Config 0	00110011 ($33)
1	Config 1	00011111 ($1F)
2	Config 2	00010000 ($10)
3	Config 3	00100000 ($20)
Status registers		
4	Status 0	00000000 ($00)
5	Status 1	-

Config 0

Bits	Name	Description
<4:0>	Wait_time	Defines the time to wait between the “LOCKED” state and “READY” state
<7:5>	Loss_of_lock_time	Defines the number of erroneous cycles the chip tolerates before it goes into the “OUT-OF-LOCK” state

Config 1

Bits	Name	Description
<3:0>	PLL_lock_time	Defines the time between the OUT-OF-LOCK state and the LOCKED state.
<4>	en_soft_loss_of_lock	If 1 then the GOL can tolerate a number of erroneous cycles defined by Loss_of_lock time, otherwise the state goes immediately to OUT-OF-LOCK after the first error.
<5>	en_loss_of_lock_count	If 1 then one TX_LOLC cycle is inserted between LOCKED and READY, where the number of loss-of-lock events is transmitted as a 16-bit data word.
<6>	en_force_lock	[Disables lock state machine if 1. Only used for testing/debugging.]
<7>	en_self_test	If 1, a running 16 bit counter generates transmission data

Config 2

Bits	Name	Description
<4:0>	PLL_current	Defines the charge-pump current of the internal phase-locked loop (PLL).
<6:5>	test_sel	[Selects signal to appear on test_analog pad. Only used for testing/debugging.]
<7>	en_flag	Enables flag bits in G-Link mode

Config 3

Bits	Name	Description
<6:0>	LD_current / driver_strength	Defines the bias current for the Laser Driver (LD) or the strength of the 50 Ohm line driver
<7>	use_conf_regs	When 1, the content of the Configuration registers 1 and 2 are used to define the value for PLL_current and LD_current. If 0, the values are derived from the encoded values on the pads.

Status 0

Bits	Name	Description
<7:0>	loss_of_lock_count	Number of "loss-of-lock" events since last reset.

Status 1

Bits	Name	Description
<7:6>	link_control_state_A	Current state of link initialisation logic A
<5:4>	link_control_state_B	Current state of link initialisation logic B
<3:2>	link_control_state_C	Current state of link initialisation logic C
<1>	conf_glink	Reads value on conf_glink pin. If 1, then the chip is configured on G-Link mode, otherwise in Ethernet mode.
<0>	conf_wmode16	Reads value on conf_wmode16 pin. If 1 then the chip accepts 16-bit wide data and transmits with 800 Mbit/s, otherwise 32 bit data and 1.6 Gbit/s speed.

References

[1] IEEE Std 802.3, 1998 Edition

[2] C. Yen, R. Walker, P. Petruno, C. Stout, B. Lai and W. McFarland, “G-Link: “Achipset for Gigabit-Rate Data Communication,” Hewlett-Packard Journal, Oct. 92.

[3] F. Vasey, C. Azevedo, G. Cervelli, K. Gill, R. Grabit and F. Jensen, “Optical links for the CMS Tracker,” *Proc. of the fifth workshop on electronics for LHC experiments*, pp. 175-179, Snowmass, 1999

[4] “The I2C-BUS specification”, Philips Semiconductors, Version 2.1, January 2000

[5] C. M. Maunder and R. E. Tulloss, “The Test Access Port and Boundary-Scan Architecture,” IEEE Computer Society Press, 1990

[6] K. Kloukinas, F. Faccio, A. Marchioro and P. Moreira, “Development of a radiation tolerant 2.0V standard cell library using a commercial deep submicron CMOS technology for the LHC experiments”, *Proc. of the fourth workshop on electronics for LHC experiments*, pp. 574-580, Rome, 1998

[7] J. Maneatis, “low-jitter process-independent DLL and PLL based on self-biased techniques,” *IEEE J. Solid-State Circuits*, vol. 31, pp. 1723-1732, Nov. 1996.

[8] I. Novof, J. Austin, R. Kelkar, D. Strayer and S. Wyatt, “Fully integrated CMOS phase-locked loop with 15 to 240 MHz locking range and ±50 ps jitter,” *IEEE J. Solid-State Circuits*, vol. 30, pp. 1259-1266, Nov. 1996.

[9] P. Jarron, et al, “Deep submicron CMOS technologies for the LHC experiments,” Nuclear Physics B (Proc Suppl.), vol. 78, pp. 625-634, 1999

[10] H. Nussbaumer, “Computer Communication Systems: Data Circuits Error Detection Data Links,” Vol. 1, John Wiley & Sons, 1990

[11] T. Toifl, P. Moreira and A. Marchioro, “Measurements of radiation effects on the timing, trigger and control receiver (TTCrx) ASIC,” to appear in the *Proc. of the sixth workshop on electronics for LHC experiments*

[12] F. Faccio, et al, “Single event effects in static and dynamic registers in a 0.25μm CMOS technology*,” IEEE Trans. Nucl. Sci*., vol. 46, no. 6, pp. 1434-1439, Dec. 1999

[13] P. Moreira, J. Christiansen, A. Marchioro, E. van der Bij, K. Kloukinas, M. Campbell and G. Cervelli, “A 1.25 Gbit/s Serializer for LHC Data and Trigger Optical Links”, *Proceedings of the Fifth Workshop on Electronics for LHC Experiments*, Snowmass, Colorado, USA, 20-24 September 1999, pp. 194-198

[14] M. Huhtinen and F. Faccio, “Computational method to estimate Single Event Upset rates in an accelerator environment”, *Nuclear Instruments and Methods A*, vol. 450, pp. 155-170, 2000

[15] V. Puget, D. Lewis, H Lapuyade, R. Briand, P. Fouillat, L. Sarget, M.-C. Calvet, “Validation of radiation hardened designs by pulsed laser testing and SPICE analysis,” *Microelectronics Reliability*, Vol. 39, pp. 931-935, 1999